## Exercise 1 : Step-by-Step Construction of the Simulation

### Exercise 1: Building the Model

To build the model, you need to open a new model window and place the required modules on the screen: two Create, two Assign, four Process, two Decide, three Record, and three Dispose modules. Your model window should now look something like Figure 1-1, assuming you've made the Connections or used the Auto-Connect feature (Object menu) while placing the modules in the appropriate sequence (the numbers inside your module shapes might be different if you placed your modules in a different order, but that doesn't matter, since they're all 'blanks' at this point). You might want to use the File . Save function now to save your model under a name of your choosing. Now, let's open each module and enter the information required to complete the model. Start with the Create 1 module that will create the arriving Part A entities. Display 1-1 provides the information required to complete this module. Note that this is very similar to the Create module.

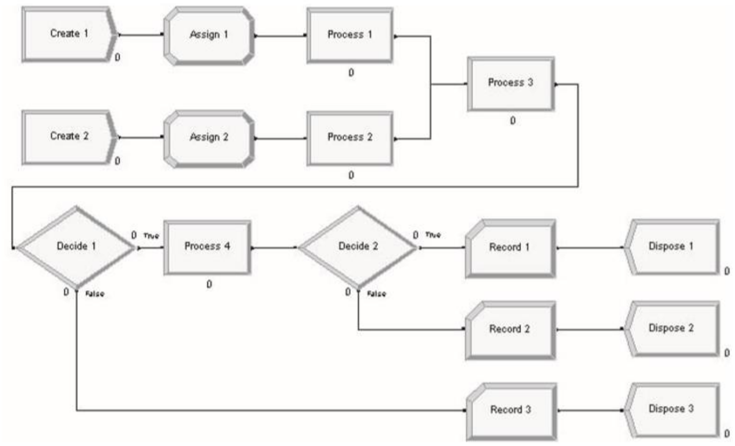
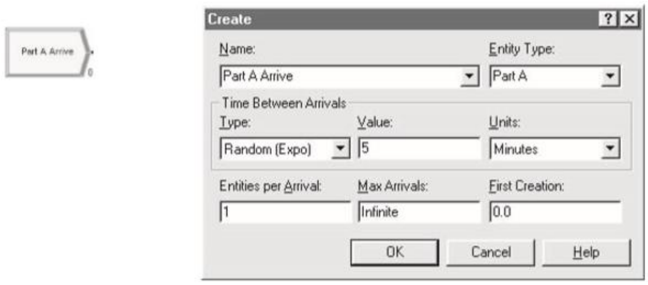


*Figure 1-1. Model Window of Placed Modules*

We've given the module a different name and specified the Entity Type as Part A. The Time Between Arrivals is Random (that is, an exponential distribution) with a Value (that is, mean) of 5, and the units are set to Minutes. The remaining entries are the default options. We can now accept the module by clicking OK.



Create

Name: Part A Arrive

Entity Type: Part A

Time Between Arrivals

Type: Random (Expo)

Value: 5

Units: Minutes

Entities per Arrival: 1

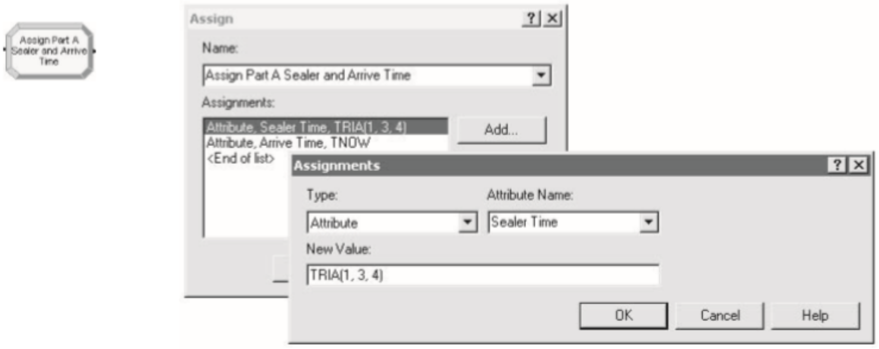
Max Arrivals: Infinite

First Creation: 0.0

| | |
|---|---|
| Name | Part A Arrive |
| Entity Type | Part A |
| Time Between Arrivals | |
| Type | Random (Expo) |
| Value | 5 |
| Units | Minutes |

*Display 1-1. The Completed Part A Create Dialogue Box*

The Create module for the Part B arrivals is very similar to that for Part A, as shown in Display 1-2 (we'll skip the graphics, since they're almost the same as what you just saw), except we have filled in one additional field (Entities per Arrival) to reflect the batch size of four. Recall that the Part B entities arrive in batches of four. Thus, this entry will cause each arrival to consist of four separate entities rather than one. Having created the arriving parts, we must next define an attribute Sealer Time and assign it a sealer processing time, which is different for each part type. We'll assign these values in the Assign 1 and Assign 2 modules that we previously placed. The Part A assignment is shown in Display 1-2. We've defined the new attribute and assigned it a value from a TRIA(1, 3, 4) distribution. We've also defined an attribute, Arrive Time, which is used to record the arrival time of the entity.



Assign

Name: Assign Part A Sealer and Arrive Time

Assignments:
Attribute, Sealer Time, TRIA(1, 3, 4)
Attribute, Arrive Time, TNOW
<End of list>

Assignments

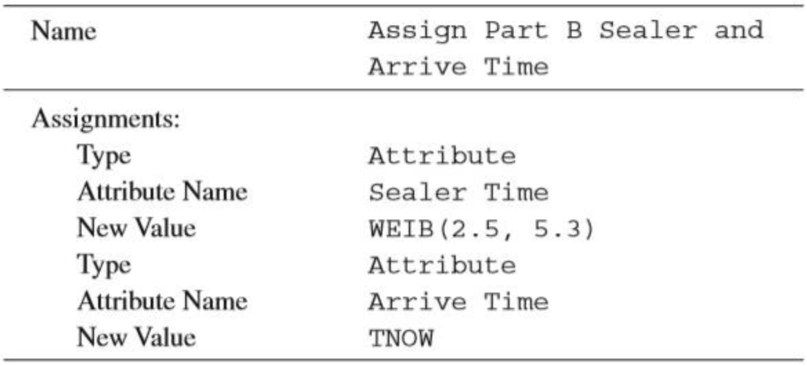
Type: Attribute

Attribute Name: Sealer Time

New Value: TRIA(1, 3, 4)

| | |
|---|---|
| Name | Assign Part A Sealer and Arrive Time |
| Assignments: | |
| Type | Attribute |
| Attribute Name | Sealer Time |
| New Value | TRIA(1, 3, 4) |
| Type | Attribute |
| Attribute Name | Arrive Time |
| New Value | TNOW |

*Display 1-2. Assigning the Part A Sealer Time and Arrival Time*



| Name | Assign Part B Sealer and Arrive Time |
|---|---|
| Assignments: | |
| Type | Attribute |
| Attribute Name | Sealer Time |
| New Value | WEIB(2.5, 5.3) |
| Type | Attribute |
| Attribute Name | Arrive Time |
| New Value | TNOW |

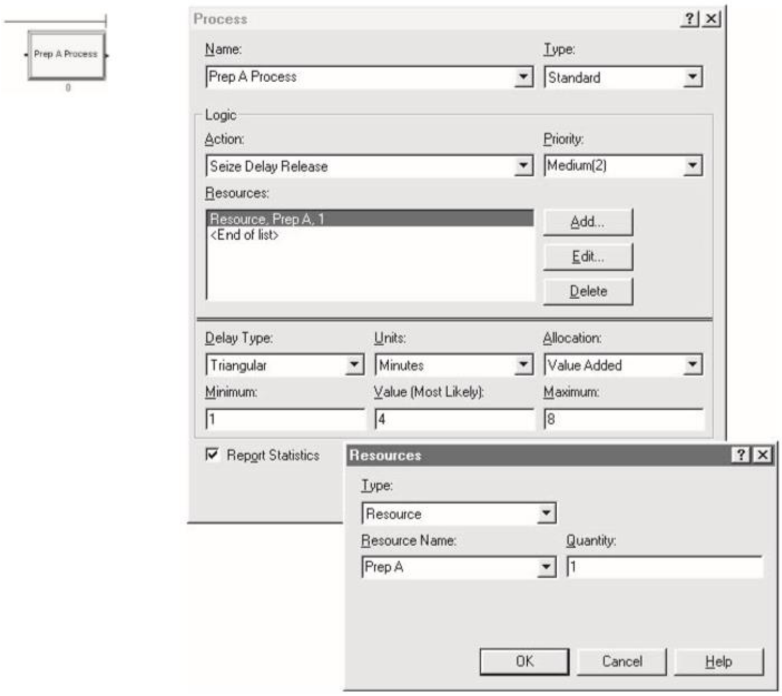
*Display 1-3. Assigning the Part B Sealer Time and Arrival Time*

The Arena variable TNOW provides the current simulation time, which in this case is the time the part arrived or was created (a good way to discover TNOW is to right-click in the New Value field of the Assign module's Assignments dialogue box, select Build Expression, guess Date and Time Functions, and it's first in the list, described as Current Simulation Time).

The assignment to the Sealer Time and Arrival Time attributes for Part B is shown in Display 1-3. Although four entities are created in the previous module for each arrival, they'll each be assigned a different (independent) value from the Sealer Time distribution in the following Assign module.

Having completed the two part-arrival modules and the assignment of the sealer times, we can now move to the two prep areas that are to be modelled using the two Process modules previously placed. The completed dialogue box for the Prep A Process area is given in Display 1-4.

The Process module has four possible Actions: Delay, Seize Delay, Seize Delay Release, and Delay Release. The Delay action will cause an entity to undergo a specified time delay. This Action does not require a Resource. This implies that waiting will occur and that multiple entities could undergo the Delay simultaneously. Since our prep area requires the use of a machine or Resource, we need an Action that will allow for waiting, queueing until the prep resource is available, and delaying for the processing time. The Seize Delay Action provides the waiting and delaying, but it does not release the Resource at the end of processing to be available for the next entity. If you use this Action, it is assumed that the Resource would be released downstream in another module. The Seize Delay Release option provides the set of Actions required to model our prep area accurately. The last Action, Delay Release, assumes that the entity that previously seized a Resource will undergo a Delay here, followed by the release of the Resource. You might notice that when you select one of the last three options, a list box appears in the empty space below the Action selection box. Click Add to enter the Resource information. When entering data, we strongly urge you to make use of drop-down lists whenever possible. The reason for this caution is that once you type a name, you must always match what you typed the first time. Arena names are not case-sensitive, but the spelling and any embedded blanks must be identical. Picking a name from the list ensures that it is the same. If you type in a slightly different name, Arena will give you an error message the first time you check or attempt to run the model (or, worse still, your model might run but it will be wrong).



Name: Prep A Process

Type: Standard

Action: Seize Delay Release

Priority: Medium(2)

Resources: Resource, Prep A, 1
<End of list>

Delay Type: Triangular

Units: Minutes

Allocation: Value Added

Minimum: 1

Value (Most Likely): 4

Maximum: 8

☑ Report Statistics

Type: Resource

Resource Name: Prep A

Quantity: 1

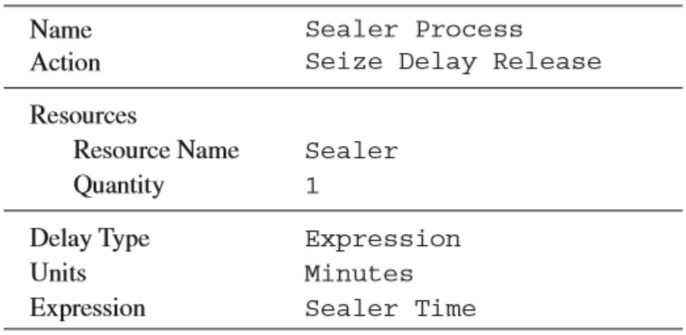
| | |
|---|---|
| Name | Prep A Process |
| Action | Seize Delay Release |
| Resources | |
| Type | Resource |
| Resource Name | Prep A |
| Quantity | 1 |
| Delay Type | Triangular |
| Units | Minutes |
| Minimum | 1 |
| Value (Most Likely) | 4 |
| Maximum | 8 |

*Display 1-4. Prep A Process Dialogue Box*

Also note that when you place a module, Arena automatically provides default names and values. These default names are the object name (module, resource, etc.) with an appended number. The appended number is incremented for each additional name, if a unique name is required; for example, Process 1, Process 2, and so on. There are two reasons for this. The first is a matter of convenience – you can accept the default resource name, or you can change it. The second reason is that all names for any objects in Arena must be unique, even if the object type is different. If the same name was used more than once, Arena could not determine which object to associate with the name. To help you, Arena does a lot of automatic naming, most of which you won't even notice. For example, if you click on the Queue data module, you'll see that Arena also assigned the name Prep A Process.Queue to the queue at this prep area. In most cases, you can assign your own names rather than accepting the default names.

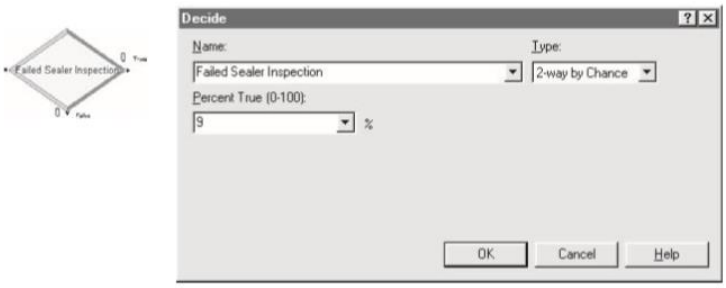
You might also notice that when you select either of the two actions that include a seize and then accept the module, Arena will automatically place an animated queue (a horizontal line with a small vertical line on the right) near the associated Process module. This will allow you to visualise entities waiting in the queue during the simulation run. If you click on this queue, the queue name will be displayed. The second Process module is filled out in an almost identical fashion, with the exception of the name (Prep B Process), the resource name (Prep B), and the parameters for the process time (3, 5, 10). We have not included a display for this module. The next step is to enter data for the Sealer operation, which is the third Process module we placed. The entries for the dialogue box are shown in Display 1-5. Note that in the upstream Assign modules we defined the attribute Sealer Time when the arriving parts were created. When an entity gains control of, or seizes, the resource, it will undergo a process delay equal to the value contained in its Sealer Time attribute.

The inspection following the sealer operation is modelled using the first Decide module. We'll accept the default Type, 2-way by Chance, as we have only a pass or fail option, and it will be decided by chance. The dialogue box requires that we enter a Percent True, and it provides two ways for entities to leave the module – True or False. In this case, we will enter the Percent True as 9%. This will result in 9% of the entities (which we'll treat as the failed items) following the True branch, and 91% (the passed items) following the False branch. Parts that pass are sent to Shipping, and parts that fail are sent to Rework.



| | |
|---|---|
| Name | Sealer Process |
| Action | Seize Delay Release |
| Resources | |
| Resource Name | Sealer |
| Quantity | 1 |
| Delay Type | Expression |
| Units | Minutes |
| Expression | Sealer Time |

*Display 1-5. The Sealer Dialogue Box*



Name: Failed Sealer Inspection

Type: 2-way by Chance

Percent True (0-100): 9 %

| | |
|---|---|
| Name | Failed Sealer Inspection |
| Percent True | 9 |

*Display 1-6. The Sealer Inspection Dialogue Box*

The data for this Decide module is shown in Display 1-6. If you've been building this model as we've moved through the material, now would be a good time to click the Save button – you never know when somebody might bump the power switch! The remaining Process module will be used to model the rework activity. The data for this Process module is shown in Display 1-7. This module is very similar to the Prep A and B Process modules, with the exceptions of the Name, Resource Name, and Expression.

The final Decide module is used to separate the salvaged and scrapped parts following rework. The data for this Decide module is shown in Display 1-8. We've chosen the True branch to represent the scrapped parts (20%) and the False branch to represent the salvaged parts.

| | |
|---|---|
| Name | Rework Process |
| Action | Seize Delay Release |
| Resources | |
| Resource Name | Rework |
| Quantity | 1 |
| Delay Type | Expression |
| Units | Minutes |
| Expression | EXPO(45) |

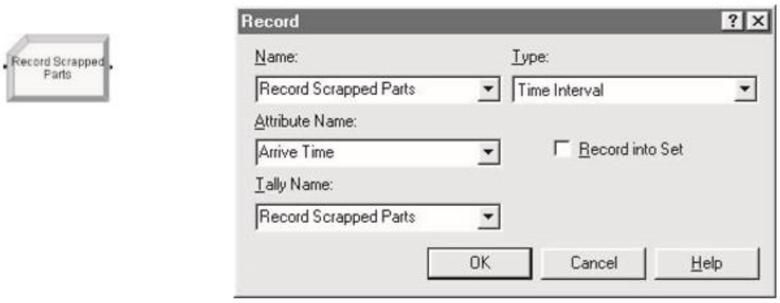
*Display 1-7. The Rework Process Dialogue Box*

| | |
|---|---|
| Name | Failed Rework Inspection |
| Percent True | 20 |

*Display 1-8. The Rework Inspection Dialogue Box*

Having defined all of the operations, we now need to fill in the Record and Dispose modules. Remember that as part of the simulation output, we wanted to collect statistics on resource utilisation, number in queue, and time in queue at each of the operations. These three statistics are automatically collected whenever you use a Process module with an Action option that requires a Resource (assuming that the Report Statistics box for the module is checked and that the Processes box is checked in Run . Setup . Project Parameters). We also wanted statistics on the cycle time separated by shipped parts, salvaged parts, and scrapped parts. The Record module provides the ability to collect these cycle times in the form of tallies. The completed dialogue box for the scrapped parts Tally is shown in Display 1-9. We picked the Type Time Interval from the drop-down list. The Tally Name defaults to the module name. This will cause Arena to record as a Tally statistic the time between the attribute carrying the arrival time (Arrive Time) of the part to the system and the time that it arrived at this Record module, which will be the entity's time in system.

The remaining two Record modules are named Record Salvaged Parts and Record Shipped Parts.  We have not included displays on these modules as they are completely analogous to the Record Scrapped Parts module in Display 1-9. The final three modules dispose of the entities as they leave the system. For this model, we could have directed all entities to a single Dispose module. However, one of the features of a Dispose module is the inclusion of an animation variable, which appears near the lower right-hand corner of the module. This animation variable will display the current count for the total number of entities that have passed through this module during the run and allow the viewer to discern the proportion of entities that have taken each of the three possible paths through the system.



Record

Name: Record Scrapped Parts

Type: Time Interval

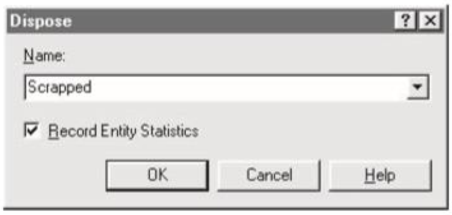
Attribute Name: Arrive Time

□ Record into Set

Tally Name: Record Scrapped Parts

| | |
|---|---|
| Name | Record Scrapped Parts |
| Type | Time Interval |
| Attribute Name | Arrive Time |
| Tally Name | Record Scrapped Parts |

*Display 1-9. The Record Scrapped Parts Dialogue Box*



Name: Scrapped

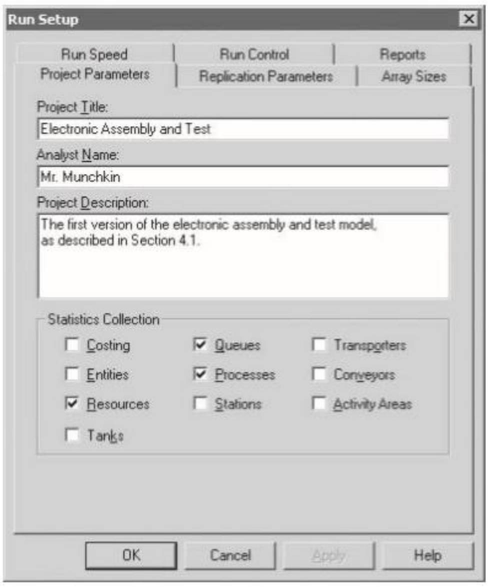
☑ Record Entity Statistics

| Name | Scrapped |
|---|---|

*Display 1-10. The Scrapped Dispose Dialogue Box*

The data for the Scrapped Dispose module is shown in Display 1-10. We have checked the box entitled Record Entity Statistics. However, if we wanted to keep entity flow statistics only on the parts that were shipped, including the salvaged parts, then we could clear this box (leaving checks in the remaining two Dispose modules). Doing so would cause only the selected parts to be included in the automatic entity-flow statistics. Of course, you need to make sure the Entities box in Run . Setup . Project Parameters is checked in order to get any of these results.

The two other Dispose modules, Salvaged and Shipped, are filled out in a similar way. You're nearly ready to run the model. Although it has taken you some time to get this far, once you get accustomed to working with Arena, you'll find that you can accomplish these steps in only a few minutes. The model could actually run at this point, but once started, it would continue to run forever, because Arena doesn't know when to stop the simulation. You can establish the run parameters for the model by selecting Run . Setup. The Run Setup dialogue box has six tabbed pages that can be used to control the simulation run. The data for the first tab, Project Parameters, is shown in Display 1-11. We've entered the project title and analyst name so they will appear on the output reports, and also a brief synopsis under Project Description for documentation purposes. In the statistics collection area, we have cleared the Entities selection, as we don't need that data for our analysis (so we won't get statistics on time in system sorted by entity type). You might try running the simulation with this box checked to see the difference in the output reports.



Project Title:
Electronic Assembly and Test

Analyst Name:
Mr. Munchkin

Project Description:
The first version of the electronic assembly and test model,
as described in Section 4.1.

Statistics Collection

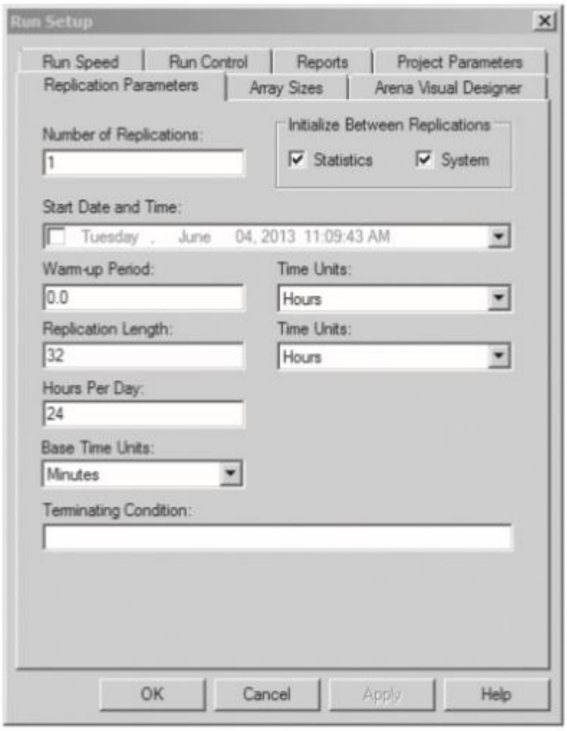
☐ Costing ☑ Queues ☐ Transporters
☐ Entities ☑ Processes ☐ Conveyors
☑ Resources ☐ Stations ☐ Activity Areas
☐ Tanks

| | |
|---|---|
| Project Title | `Electronic Assembly and Test` |
| Analyst Name | `Mr. Munchkin` |
| Project Description | `The first version of the electronic assembly and test model, as described in Section 4.1.` |
| Statistics Collection | |
| Entities | *clear* |

*Display 1-11. The Run Setup Project Parameters*

You also need to specify the run length, which is done under the Replication Parameters tab. We've set the Replication Length to 32 hours (4 consecutive 8-hour shifts), the Base Time Units to Minutes, and kept the default settings for the remaining fields. The completed dialogue box is shown in Display 1-12. We've also accepted the default settings for the remaining four tabs in Run . Setup: Array Sizes, Run Speed, Run Control, and Reports. You might want to look at these tabs to get an idea of the options available.

Before we run our newly created model, let's give it one final tweak. Since we have two different part types, it would be useful to distinguish between them in the animation. Click on the Entity data module found in the Basic Process panel and note that the initial picture for both parts is Picture.Report. When we run our model, all of our parts will use this same icon for displaying entities on the animation. Now, click on the Initial Picture cell for Part A and use the list to select a different picture. We've chosen the blue ball for Part A and the red ball for Part B, as shown in Display 1-13.



Number of Replications: 1

Initialize Between Replications: ☑ Statistics ☑ System

Start Date and Time: □ Tuesday , June 04, 2013 11:09:43 AM

Warm-up Period: 0.0 Time Units: Hours

Replication Length: 32 Time Units: Hours

Hours Per Day: 24

Base Time Units: Minutes

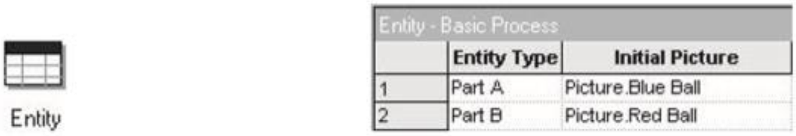
Terminating Condition:

| | |
|---|---|
| Replication Length | 32 |
| Base Time Units | Minutes |

*Display 1-12. The Run Setup Replication Parameters*

This will allow us to distinguish easily between the two parts in the animation. If you're interested in seeing what these icons look like, you can select Edit . Entity Pictures from the menu bar at the top of your screen to open the Entity Picture Placement window, which will allow you to see the icons currently available down the left column. We'll show you later how to use this feature in more detail.

Your final model should look something like Figure 1-2.



| | Entity Type | Initial Picture |
|---|---|---|
| 1 | Part A | Picture.Blue Ball |
| 2 | Part B | Picture.Red Ball |

| | |
|---|---|
| Initial Picture (Part A) | Picture.Blue Ball |
| Initial Picture (Part B) | Picture.Red Ball |

*Display 1-13. The Entity Data Module*

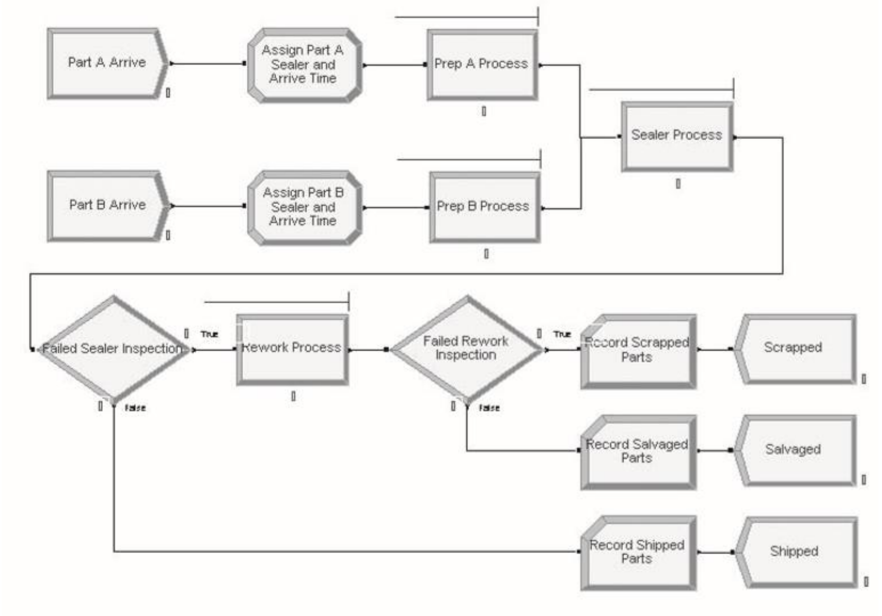


*Figure 1-2. The Final Model 1*